

# Instrucciones para Compilación y Ejecución

## Ejecución del Servidor UDP

- Clase a Ejecutar: `py.una.server.udp.UDPServer` (Puede ejecutarse en una consola aparte o desde el IDE).
- Solo puede ejecutar uno a la vez.

## Ejecución del Cliente UDP

- Clase a Ejecutar: `py.una.server.udp.UDPClient` (Puede ejecutar en una consola aparte o desde el IDE)
- Puede ejecutar varios clientes en simultáneo.

## Compilar y ejecutar vía consola

Las clases de Servidor UDP y Cliente UDP ya están definidas en el archivo pom.xml tanto del directorio servidor como del cliente. Para construir y ejecutar el proyecto se debe:

- 1) Navegar al directorio del proyecto que se desea construir (cliente o servidor):
  - `cd /ruta/a/servidor`    ó    `cd /ruta/a/cliente`
- 2) Ejecutar el siguiente comando para construir el proyecto:
  - `mvn clean package`
- 3) Al finalizar la construcción, Maven indicará el archivo .jar que se generó dentro del directorio target.
- 4) Navegar al directorio target:
  - `cd /target`
- 5) Ejecuta el archivo .jar generado:
  - `java -jar socket-0.0.1-SNAPSHOT-jar-with-dependencies.jar`

### Una vez ejecuta el comando:

El **Servidor UDP** se mantiene a la escucha de nuevos paquetes de datos en un bucle infinito, utilizando un socket de tipo *DatagramSocket*. Recibe los mensajes de los clientes y los reenvía a otros clientes que se encuentren conectados

- Los clientes son almacenados en un *ConcurrentHashMap* donde la clave es una combinación del nombre del usuario y su dirección IP, y el valor es un *InetSocketAddress*, que contiene la información de la dirección y el puerto de cada cliente. Esto permite enviar mensajes a todos los clientes conectados menos al que envía el mensaje.
- Al recibir un paquete de datos, el servidor identifica al remitente y lo procesa. Si el mensaje contiene el identificador NUEVO USUARIO, se registra el cliente. Si no, el mensaje se reenvía a todos los demás clientes.

El **Cliente UDP**, tras ingresar el nombre de usuario, envía esta información al servidor para ser registrado.

- La recepción de mensajes del servidor y envíos de mensajes ocurren en hilos separados, lo cual permite que el cliente siga enviando mensajes mientras recibe otros. Este hilo dedicado a la recepción de mensajes utiliza *DatagramPacket* para escuchar en el socket y mostrar los mensajes en la consola del cliente.
- El cliente ingresa mensajes de la consola y los envía al servidor. Estos mensajes son re-enviados a los demás clientes conectados por el servidor.
- El cliente puede salir del chat si ingresa por consola el mensaje “exit”

## Imágenes de Resultado Esperado

Para el caso del **ServidorUDP**, se visualizará por consola el siguiente mensaje indicando que el servidor ya está en funcionamiento:

```

C:\lab-socket\SDT3\servidor\target> java -jar socket-0.0.1-SNAPSHOT-jar-with-dependencies.jar
es.jar
Servidor de Chat UDP en funcionamiento...

```

Para el caso del **Cliente UDP**, se visualizará por consola el siguiente mensaje indicando que se ingrese el nombre del cliente:

```

C:\lab-socket\SDT3\cliente\target> java -jar socket-0.0.1-SNAPSHOT-jar-with-dependencies.jar
s.jar
Ingrese su nombre de usuario: 

```

Una vez ingresado el nombre del Usuario, en el servidor se visualiza el siguiente mensaje indicando que se han conectado nuevos clientes:

```

Servidor de Chat UDP en funcionamiento...

Mensaje recibido; NUEVO_USUARIO:Juan Andres
De : /127.0.0.1:53984
Clientes conectados: [Juan Andres@127.0.0.1:53984]

Mensaje recibido; NUEVO_USUARIO:Jonathan
De : /127.0.0.1:53983
Clientes conectados: [Jonathan@127.0.0.1:53983, Juan Andres@127.0.0.1:53984]

Mensaje recibido; NUEVO_USUARIO:Ivan
De : /127.0.0.1:52582
Clientes conectados: [Jonathan@127.0.0.1:53983, Ivan@127.0.0.1:52582, Juan Andres@127.0.0.1:53984]

```

Si el **cliente** envía un mensaje, dicho mensaje se le reenviará a todos los clientes conectados en ese momento. En dicho mensaje, se detalla el nombre del usuario que envía dicho mensaje

<pre> Ingrese su nombre de usuario: Ivan Intentando conectar a = localhost/127.0.0.1:9876 via UDP... Mensaje: Probando envio de mensaje Mensaje:  </pre>	<pre> Ingrese su nombre de usuario: Jonathan Intentando conectar a = localhost/127.0.0.1:9876 via UDP... Mensaje: Ivan: Probando envio de mensaje </pre>	<pre> Ingrese su nombre de usuario: Juan Andres Intentando conectar a = localhost/127.0.0.1:9876 via UDP... Mensaje: Ivan: Probando envio de mensaje </pre>
--	--	---

En el **servidor** también se visualiza dicho mensaje

```

Mensaje recibido: Ivan:Probando envio de mensaje
De : /127.0.0.1:52582
Clientes conectados: [Jonathan@127.0.0.1:53983, Ivan@127.0.0.1:52582, Juan Andres@127.0.0.1:53984]

```

Se finaliza el proceso del cliente una vez ingrese el mensaje “**exit**”

<pre>Ingrese su nombre de usuario: Ivan Intentando conectar a = localhost/127.0.0.1:9876 via UDP... Mensaje: Probando envio de mensaje Mensaje: Jonathan: Test de envio 1 Juan Andres: Test de envio 2 █</pre>	<pre>Ingrese su nombre de usuario: Jonathan Intentando conectar a = localhost/127.0.0.1:9876 via UDP... Mensaje: Ivan: Probando envio de mensaje Test de envio 1 Mensaje: Juan Andres: Test de envio 2 █</pre>	<pre>Intentando conectar a = localhost/127.0.0.1:9876 via UDP... Mensaje: Ivan: Probando envio de mensaje Test de envio 2 Mensaje: Jonathan: Test de envio 1 exit \lab- o ocket\SDT3\cliente\target&gt; █</pre>
--	--	---